Direct Message Passing for Hybrid Bayesian Networks and Performance Analysis

Wei Sun and KC Chang

The Sensor Fusion Lab, Center of Excellence in C4I
Department of Systems Engineering and Operations Research
George Mason University
Fairfax, VA 22030

ABSTRACT

Probabilistic inference for hybrid Bayesian networks, which involves both discrete and continuous variables, has been an important research topic over the recent years. This is not only because a number of efficient inference algorithms have been developed and used maturely for simple types of networks such as pure discrete model, but also for the practical needs that continuous variables are inevitable in modeling complex systems. Pearl's message passing algorithm provides a simple framework to compute posterior distribution by propagating messages between nodes and can provides exact answer for polytree models with pure discrete or continuous variables. In addition, applying Pearl's message passing to network with loops usually converges and results in good approximation. However, for hybrid model, there is a need of a general message passing algorithm between different types of variables. In this paper, we develop a method called Direct Message Passing (DMP) for exchanging messages between discrete and continuous variables. Based on Pearl's algorithm, we derive formulae to compute messages for variables in various dependence relationships encoded in conditional probability distributions. Mixture of Gaussian is used to represent continuous messages, with the number of mixture components up to the size of the joint state space of all discrete parents. For polytree Conditional Linear Gaussian (CLG) Bayesian network, DMP has the same computational requirements and can provide exact solution as the one obtained by the Junction Tree (JT) algorithm. However, while JT can only work for the CLG model, DMP can be applied for general nonlinear, non-Gaussian hybrid model to produce approximate solution using unscented transformation and loopy propagation. Furthermore, we can scale the algorithm by restricting the number of mixture components in the messages. Empirically, we found that the approximation errors are relatively small especially for nodes that are far away from the discrete parent nodes. Numerical simulations show encouraging results.

Keywords: Hybrid Bayesian network, Gaussian mixture, message passing

1. INTRODUCTION

Bayesian network (BN)[1234] is a directed acyclic graph (DAG) consisting of nodes and arrows, in which node represents random variables, and arrow represent probabilistic dependence relationship. Each node in BN has a pre-defined conditional probability distribution (CPD) that parameterizes the model. BN has been a powerful probabilistic model for decision support under uncertainty over a few decades. Even though both exact and approximate inference for BN are NP-hard in general[5],[6] A number of inference algorithms has been developed in the literature.[7] However, inference for hybrid models with both discrete and continuous variables has many difficulties and open issues. The simplest hybrid Bayesian network model is called Conditional Linear Gaussian (CLG),[8] in which a discrete node can have continuous children, but a continuous node is not allowed to have discrete child. Given its discrete parents, a continuous variable in CLG has a linear functional relationship with its continuous parents. The Junction Tree algorithm[9810] can be applied for CLG to provide a solution with the exact first two moments of posterior distributions for hidden continuous variables, and exact posterior probabilities for hidden discrete variables. But the complexity of the Junction Tree algorithm is exponential to the size of the largest clique from the strongly triangulated graph, which is determined by the size of state space of all discrete parent nodes in CLG.[11] For a general hybrid BN with nonlinear and/or non-Gaussian model, as

Send correspondence to Wei Sun: wsun@gmu.edu, Telephone: 1 703 993 5538

Signal Processing, Sensor Fusion, and Target Recognition XIX, edited by Ivan Kadar,
Proc. of SPIE Vol. 7697, 76970S · © 2010 SPIE · CCC code: 0277-786X/10/$18 · doi: 10.1117/12.852088

commonly known, there is no existing method that could produce exact posterior solutions. Hence, one has to rely on approximate methods in that case.

Because of the heterogeneity of variables and arbitrary functional relationships in general hybrid model, approximate methods usually produce errors due to functional transformation, distribution approximation, discretization, or structure simplification. For example, Likelihood Weighting (LW)[12, 13] is a general-purpose simulation algorithm that is model-independent, but has difficulty handling unlikely evidence; The Mixture of Truncated Exponential (MTE) algorithm[14] produces closed form solutions, but introduces errors due to approximating arbitrary distributions using mixtures of truncated exponentials; Hybrid Loopy Propagation algorithm[15] uses mixture of Gaussian to represent continuous messages, and then computes messages by numerical integrations.

In this paper, we are particularly interested in the message passing framework because of its simplicity of implementation and good empirical performance. In[16],[17] we presented a hybrid loopy propagation method that uses network partitioning, and then integrates information via the interface nodes. This hybrid message passing algorithm also allows us to apply any suitable algorithm in individual sub-network. An advantage of the hybrid algorithm is that for each sub-network, it is easier to find an efficient algorithm for inference because of the homogeneity of variable type within the sub-network. However, the disadvantage is that we have to conduct inference conditioning on all the discrete parent nodes (i.e., interface nodes). Therefore, the algorithm has an exponential complexity proportional to the product of sizes of discrete parent nodes, which is identical to the Junction Tree algorithm.

Pearl's message passing algorithm[2] is the first exact inference method originally proposed only for polytree discrete Bayesian networks. For networks with loops, applying Pearl's message passing in a "loopy" fashion usually converges and provides good approximate results.[18] For pure continuous networks, similarly, Pearl's algorithm is applicable with continuous message represented in appropriate forms such as the first two moments of Gaussian. However, it lacks an efficient way to exchange messages between discrete and continuous variables in a hybrid network. In this paper, we attempt to unify the message passing framework for general hybrid networks without network partitioning or graph transformation. Based on the traditional Pearl's message passing mechanism, we derive formulae for computing direct messages under various circumstances. A similar approach has been taken in[19] using Gaussian mixtures to approximate hybrid conditional densities. In our method, we use the original CPDs without approximation and provide exact solution for polytree networks.

The remainder of this paper is organized as follows. Section 2 describes the DMP algorithm in detail. Section 3 shows our numerical simulation examples. Finally, we summarize and discuss the future research in Section 4.

2. DIRECT MESSAGE PASSING

This section describes Direct Message Passing (DMP) algorithm in details. We first briefly review the original Pearl's message passing algorithm. We then extend it for general hybrid models.

2.1 Pearl's Message Passing Algorithm

Recall that in a polytree network, any node X d–separates evidence into $\{\mathbf{e}^+, \mathbf{e}^-\}$, where $\mathbf{e}^+$ and $\mathbf{e}^-$ are evidence from the sub-network "above" X and "below" X respectively. Every node in the network maintains two values called λ and π. λ value of X is the likelihood, defined as:

$$\lambda(X) = P(\mathbf{e}_X^-|X) \tag{1}$$

And π value of X, defined as:

$$\pi(X) = P(X|\ \mathbf{e}_X^+) \tag{2}$$

is the conditional probability distribution of X given $\mathbf{e}_X^+$. It is easy to see that the belief of a node X given all evidence is just the normalized product of λ and π values:

$$BEL(X) \quad = \quad P(X|\mathbf{e}) = P(X|\mathbf{e}_X^+, \mathbf{e}_X^-)$$

$$
\begin{aligned}
&= \frac{P(\mathbf{e}_X^-|X,\mathbf{e}_X^+)P(X|\mathbf{e}_X^+)P(\mathbf{e}_X^+)}{P(\mathbf{e}_X^+,\mathbf{e}_X^-)} \\
&= \alpha P(\mathbf{e}_X^-|X)P(X|\mathbf{e}_X^+) \\
&= \alpha\lambda(X)\pi(X) \qquad (3)
\end{aligned}
$$

where α is a normalizing constant. In message passing, every node sends message to each of its parents and each of its children. Based on received messages, every node updates its λ and π values correspondingly. The general message propagation equations of Pearl's algorithm are the following:[2]

$$
\pi(X) = \sum_{\mathbf{T}} P(X|\ \mathbf{T}) \prod_{i=1}^{m} \pi_X(T_i) \qquad (4)
$$

$$
\lambda(X) = \prod_{j=1}^{n} \lambda_{Y_j}(X) \qquad (5)
$$

$$
\pi_{Y_j}(X) = \alpha \left[\prod_{k \neq j} \lambda_{Y_k}(X) \right] \pi(X) \qquad (6)
$$

$$
\lambda_X(T_i) = \sum_X \lambda(X) \sum_{T_k:\ k \neq i} P(X|\mathbf{T}) \prod_{k \neq i} \pi_X(T_k) \qquad (7)
$$

where $T(T_1, T_2, ..., T_n)$ are parents of node X; $Y(Y_1, Y_2, ..., Y_m)$ are children of node X; $\lambda_{Y_j}(X)$ is the λ message node X receives from its child Y_j, $\lambda_X(T_i)$ is the λ message X sends to its parent T_i; $\pi_X(T_i)$ is the π message node X receives from its parent T_i, $\pi_{Y_j}(X)$ is the π message X sends to its child Y_j; and α is a normalizing constant.

Equations 4 to 7 are recursive equations, so we need to initialize messages properly to start the message propagation. Again, Pearl's algorithm is originally designed for discrete polytree networks, so these propagation equations are for computing discrete probabilities. When Pearl's algorithm applies to pure discrete polytree network, the messages propagated are exact and so are the beliefs of all nodes after receiving all messages. For pure continuous networks with arbitrary distributions, we proposed a method called Unscented Message Passing[20] using a similar framework with different message representations and a new corresponding computation method. However, with both discrete and continuous variables in the model, passing messages directly between different types of variables requires additional techniques.

2.2 Direct Message Passing between Discrete and Continuous Variables

In typical CLG, continuous node can not have discrete child. Therefore, the only case we need to be concerned for message exchanging between different types of variables is when a continuous node has discrete parents. Without loss of generality, suppose that we have a typical hybrid network involving a continuous node X with a discrete parent node D and a continuous parent node U. As shown in Figure 1, messages sent between these nodes are: (1) π message from D to X, denoted as $\pi_X(D)$; (2) π message from U to X, denoted as $\pi_X(U)$; (3) λ message from X to D, denoted as $\lambda_X(D)$; and (4) λ message from X to U, denoted as $\lambda_X(U)$. In addition, each node needs to maintain its λ and π values.

Let us look at these messages one by one, and derive their corresponding formula based on the traditional Pearl's message passing mechanism. First, recall from Equation 6, $\pi_X(D)$ can be computed by substitution:

$$
\pi_X(D) = \alpha \left[\prod_{child \neq X} \lambda_{child}(D) \right] \pi(D) \qquad (8)
$$

where $\lambda_{child}(D)$ is λ message sent to D from each of its children except X, and $\pi(D)$ is the message sent from the discrete sub-network "above" D that can be easily computed. Note that $\lambda_{child}(D)$ is always in the form of discrete vector. After normalizing, $\pi_X(D)$ is a discrete probability distribution serving as the mixing prior for a Gaussian mixture.

Similarly, but in a different form, $\pi_X(U)$ can be computed as below,

$$\pi_X(U) = \alpha \left[\prod_{child \neq X} \lambda_{child}(U) \right] \pi(U) \tag{9}$$

where $\lambda_{child}(U)$ are λ messages sent to U from its continuous children other than X. These λ messages are continuous messages in the form of Gaussian mixtures. $\pi(U)$ is π value of U, and its computation depends on the type of parent nodes it has. The generalized computation of $\pi(X)$ will be described in the next paragraph. Finally, the resulting $\pi_X(U)$ is a normalized product of Gaussian mixtures, resulting in another Gaussian mixture with a greater number of components.

Now for $\pi(X)$, by applying Equation 4 with integral replacing summation for continuous variable, we have,

$$\begin{aligned} \pi(X) &= \sum_D \int_U P(X|D,U)\pi_X(D)\pi_X(U)dU \\ &= \sum_D \left[\pi_X(D) \int_U P(X|D,U)\pi_X(U)dU \right] \end{aligned} \tag{10}$$

where $\pi_X(D)$ and $\pi_X(U)$ are π messages sent from D and U respectively. For a given $D = d$, $P(X|D = d, U)$ defines a probabilistic functional relationship between X and its continuous parent U. The integral of $P(X|D = d, U)\pi_X(U)$ over U is equivalent to a functional transformation of $\pi_X(U)$, which is a continuous message in the form of a Gaussian mixture. In this functional transformation process, we pass each Gaussian component individually to form a new Gaussian mixture. Essentially, $\pi(X)$ is a mixture of continuous distributions weighted by $\pi_X(D)$. To avoid potential growing complexity of the message, it is possible to collapse it into a single Gaussian density or a Gaussian mixture with fewer components as an approximation to reduce the computational requirements.

$\lambda(X)$ is relatively straightforward to compute as it is the product of λ messages from each of its children, which must be continuous variables due to the CLG model restriction. However, since we represent a continuous message as a Gaussian mixture, the product of a set of Gaussian mixtures will be another Gaussian mixture with increased number of components.

Let us now turn to the computation of messages sent from X to its parents D and U. As shown in Equation 7, λ message sending to its parents is essentially an inverse functional transformation of the product of λ value of the node itself and π messages sending from all of its other parents via the function defined in the CPD of X. It can be derived as,

$$\lambda_X(D = d) = \int_X \lambda(X) \left[\int_U P(X|D = d, U)\pi_X(U)dU \right] dX \tag{11}$$

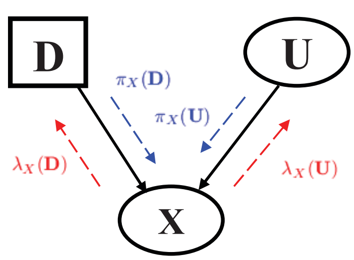


Figure 1. A typical node with hybrid CPD — continuous node X has discrete parent D and continuous parent U.

where $\int_U P(X|D=d,U)\pi_X(U)dU$ is a functional transformation of a distribution over U to X, resulting in a probability distribution of X. Further, by multiplying with $\lambda(X)$ and integrating over X, it ends up with a non-negative constant, serving as a likelihood of X given $D=d$.

Similarly, λ message sending from X to its continuous parent U can be computed as:

$$\begin{aligned}\lambda_X(U) &= \int_X \lambda(X) \sum_D P(X|D,U)\pi_X(D)\pi_X(D)dX \\ &= \sum_D \left[\pi_X(D) \int_X \lambda(X)P(X|D,U)dX\right] \end{aligned} \tag{12}$$

Note that $\int_X \lambda(X)P(X|D,U)dX$ is an integral of product of X's λ value and its conditional probability distribution and this integral is over X itself. Therefore it results in a density estimate of its parent multiplied by a coefficient. This coefficient is very critical in computing mixing priors with $pi_X(D)$ when there is more than one component in the mixture distributions.

Equations 8 to 12 form a baseline for computing messages between discrete and continuous variables. Along with the well-defined formulae for computing messages between the same types of variables, they together provide a general message passing framework for hybrid Bayesian network models. Note that the presence of discrete parents for continuous variable makes the corresponding continuous messages necessarily a mixture distribution. Unfortunately, the number of mixture components in the message increases exponentially with the size of joint state space of the discrete parents. In order to scale the algorithm, one alternative is to combine or reduce the mixture components into smaller ones, trading off complexity against accuracy.

2.3 Scalability Issues

In message passing process, whenever a node has a discrete parent, the size of the messages to be exchanged will increase. Particularly, the number of mixture components will increase proportionally to the size of discrete parent's state space. In particular, as shown in Equaion 10 and 12, $\pi(X)$ and $\lambda_X(U)$ represented by mixture of Gaussians have number of components equal to the size of state space of its discrete parent D. When it propagates the message to another continuous node with discrete parent, the message size will increase again accordingly. In practice, if there are many discrete parents, the number of mixture components could be large that makes the algorithm very computationally extensive. One way to resolve this issue is to combine the Gaussian mixture into a single Gaussian or a smaller Gaussian mixture with fewer components. Approximation error can be quantified in the form of Integrated Square Error (ISE) when approximating Gaussian mixture by one with reduced components.[21,22] However, it is non-trivial to estimate the inference error after the messages are compressed and propagated. In the next section, we will provide some performance results with numerical experiments to show the approximation errors empirically.

3. NUMERICAL EXPERIMENTS

Theoretically, DMP can provide exact results for a polytree CLG. For verification purpose, an example model called *Poly12CLG* as shown in Figure 2, was used for the experiment.

Assume evidence is observed on leaf nodes E, and Z. With random observations, we conducted more than 30 independent experiments and compared DMP results with the ones obtained by Junction Tree algorithm (we used the well-known commercial software Hugin as the JT implementing tool). The latter algorithm is considered to be the gold standard and the resulting solutions are served as the ground truth. All experiment results show that DMP provides the exact same answers as the ground truth.

We also conducted scalability test of DMP algorithm using the same example model *Poly12CLG*. For most decision support models, the variables of interest are usually discrete, such as entity classifications and situation hypothesis. In our experiments, we will show how the assessments of hidden discrete nodes in a CLG are being affected after collapsing Gaussian mixture into a single Gaussian when passing messages. We use absolute probability errors between two discrete distributions as the metric to evaluate the performance. In general, when a node of interest is relatively far away from the evidence, its posterior distribution would not deviate much

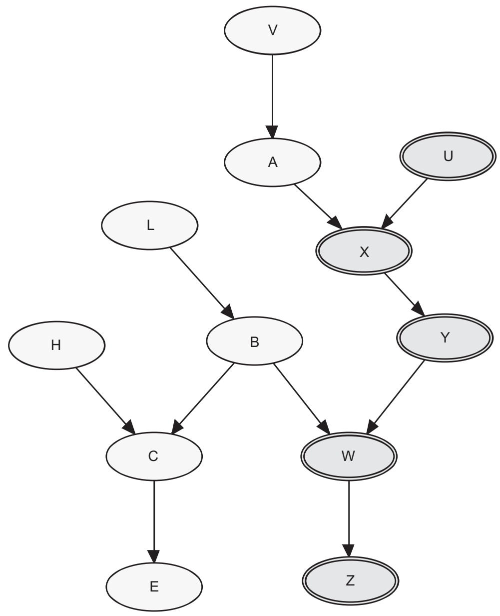


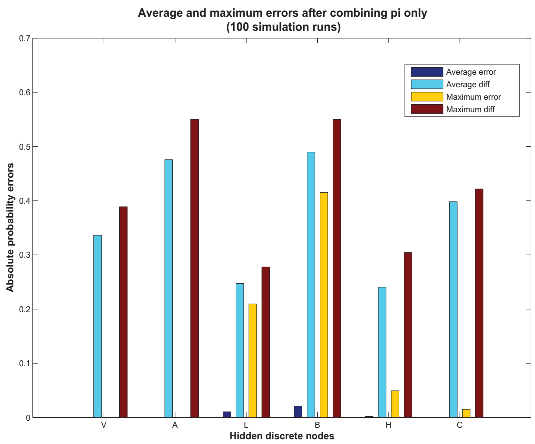
Figure 2. An example polytree CLG model called Poly12CLG, consisting of 7 discrete nodes V, A, L, B, H, C, E and 5 continuous nodes U, X, Y, W, Z.

from its prior. In that case, it is difficult to show the impact of the approximation on the inference error. So we purposely designed CPDs in *Poly12CLG* to move the true posterior probabilites away from its prior. Figure 3 shows the average and maximum errors of the approximate posterior probabilities for hidden discrete nodes V, A, L, B, H, and C, obtained after collapsing Gaussian mixtures over 100 Monte Carlo simulations. Average and maximum difference between the true posteriors and the priors over these 100 simulations are also shown in the figure for comparison. Figure 3(a) presents the estimate errors when combining π values only; Figure 3(b) shows the performance when combining λ messages only; and Figure 3(c) displays the inference errors when combining both π values and λ messages whenever mixture of Gaussian is present.

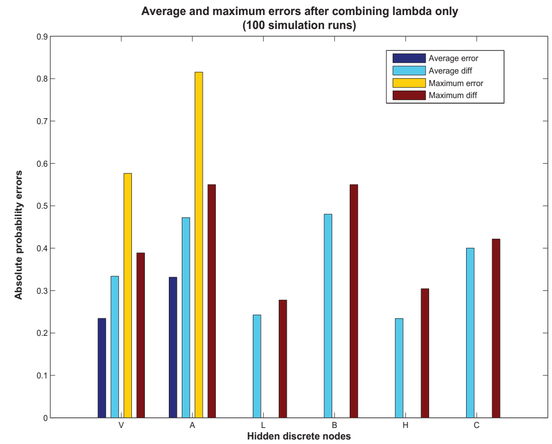
Combining π value of a node does not affect the network "above" it because π message is being sent downward in the network. Similarly, since λ message is being sent upward, combining λ message will not affect the network "below" the node. For example in Figure 3(a), the posterior probabilities of V and A are exact, and in Figure 3(b), the estimates of L, B, H, and C are also exact without inference error. When combining both π values and λ messages into one single Gaussian, all posterior distributions are not exact any more. As shown in Figure 3(c), empirical results suggest that the approximation errors diminish while the nodes are away from discrete parents. For example, the approximate estimates for nodes L, H, and C are accurate in *Poly12CLG*. However, the discrete parent nodes such as A, and B, are affected significantly. Simulation results also show that combining λ message has bigger influence on performance than combining π values.

For network with loops, another example model called *Loop13CLG* (extended from *Poly12CLG*), shown in Figure 4, was used for numerical testing. Again, we assume that leaf nodes E and Z are observable evidence nodes. With random observations, Figure 5 shows the average and maximum absolute errors of posterior probabilities for hidden discrete nodes over 100 Monte Carlo simulations. All simulation runs converge in about 11 iterations. As can be seen from the figure, average approximation errors caused by loopy propagation range from less than 1% to about 5% for hidden discrete nodes.

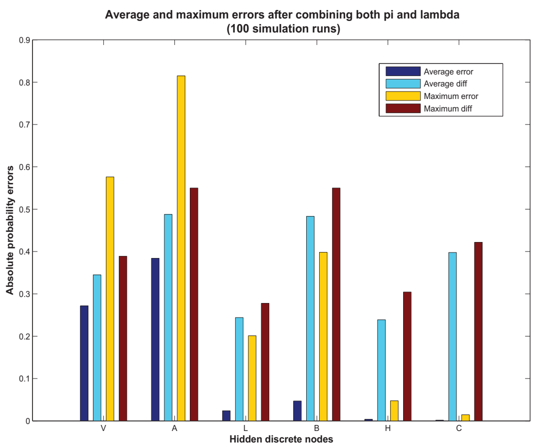
We also tested DMP with some other networks with randomly pre-defined CPDs. All simulation results suggest that the estimation errors reduce significantly as the node is further away from the discrete parent nodes.



(a) combining π values only



(b) combining λ messages only



(c) combining both π values and λ messages

Figure 3. Scalability test – performance loss after combining Gaussian mixture into one single Gaussian.

4. SUMMARY

In this paper, we present a new representation and inference algorithm called Direct Message Passing (DMP) to exchange messages between discrete and continuous variables. Essentially, DMP provides an alternative for probabilistic inference in hybrid Bayesian networks. It can be extended with unscented transformation[20] for general hybrid model with nonlinear and/or non-Gaussian distributions. Since DMP is a distributed algorithm utilizing only local information, there is no need to transform the network structure as so required by the Junction Tree algorithm. In addition, DMP does not require prior knowledge of the global network topology which could be changing dynamically. This is a major advantage of the algorithm and is particularly important to ensure scalable and reliable message exchanges in a large information network where computations can be done locally.

As shown in the empirical simulation results, DMP is scalable with a tradeoff of losing some accuracy. For decision support systems, we are mainly interested in hidden discrete variables such as entity classifications or high level situation hypothesis. The experiment results show that the estimation errors of the hidden discrete variables depend on the network topology and are somewhat bounded. Theoretically, it is non-trivial to estimate the performance bounds quantitatively due to message compressing and propagation. This points to an potential topic for future research.

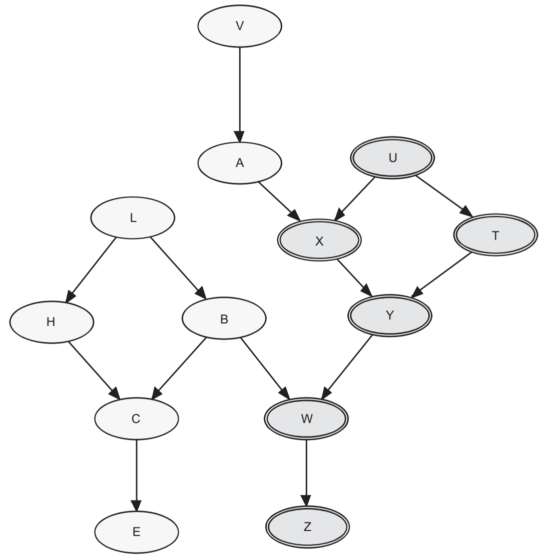


Figure 4. Loop13CLG – an example CLG model with multiple loops, consisting of 7 discrete nodes V, A, L, B, H, C, E and 6 continuous nodes U, X, TY, W, Z.

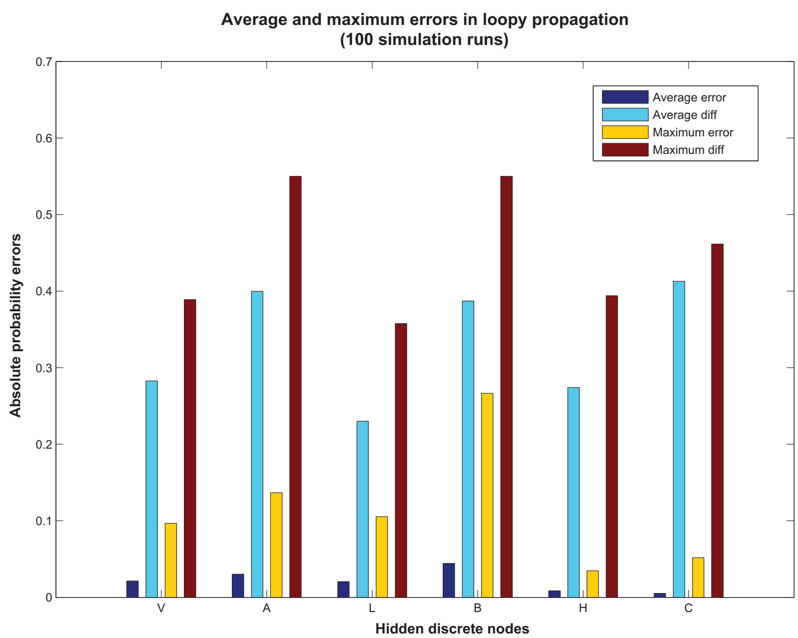


Figure 5. Performance test with loopy CLG model.

REFERENCES

[1] Charniak, E., "Bayesian networks without tears: making bayesian networks more accessible to the probabilistically unsophisticated," *AI Magazine* **12**(4), 50–63 (1991).

[2] Pearl, J., [*Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*], Morgan Kauffman, San Mateo (1988).

[3] Jensen, F., [*An Introduction to Bayesian Networks*], Springer-Verlag, New York (1996).

[4] Neapolitan, R., [*Probabilistic Reasoning in Expert Systems*], John Wiley & Sons, New York (1990).

[5] Cooper, G., "The computational complexity of probabilistic inference using bayesian belief networks," *Artificial Intelligence* **42**, 393–405 (1990).

[6] Dagum, P. and Luby, M., "Approximating probabilistic inference in bayesian belief networks is np–hard," *Artificial Intelligence* **60**, 141–153 (1993).

[7] Guo, H. and Hsu, W., "A survey of algorithms for real-time bayesian network inference," in [*AAAI/KDD/UAI–2002 Joint Workshop on Real-Time Decision Support and Diagnosis Systems*], (2002).

[8] Lauritzen, S., "Propagation of probabilities, means amd variances in mixed graphical association models," *JASA* **87**(420), 1098–1108 (1992).

[9] Lauritzen, S. and Spiegelhalter, D., "Local computations with probabilities on graphical structures and their applications to expert systems," in [*Proceedings of the Royal Statistical Society, Series B.*], **50**, 157–224 (1988).

[10] Lauritzen, S. L. and Jensen, F., "Stable local computations with conditional gaussian distributions," *Statistics and Computing* **11**, 191–203 (April 2001).

[11] Lerner, U. N., [*Hybrid Bayesian Networks for Reasoning about Complex Systems*], Stanford University (October 2002).

[12] Fung, R. and Chang, K. C., "Weighting and integrating evidence for stochastic simulation in bayesian networks," (1989).

[13] Shachter, R. D. and Peot, M. A., "Simulation approaches to general probabilistic inference on belief networks," in [*Proceedings of the Conference on Uncertainty in Artificial Intelligence*], **5** (1999).

[14] Cobb, B. R. and Shenoy, P. P., "Inference in hybrid bayesian networks with mixtures of truncated exponentials," *International Journal of Approximate Reasoning* **41**, 257–286 (April 2006).

[15] Yuan, C. and Druzdzel, M. J., "Hybrid loopy belief propagation," in [*Proceedings of the third European Workshop on Probabilistic Graphical Models*], 317–324 (2006).

[16] Sun, W. and Chang, K. C., "Hybrid message passing for general mixed bayesian networks," in [*Proceedings of the 10th International Conference on Information Fusion*], (2007).

[17] Sun, W. and Chang, K., "Message passing for hybrid bayesian networks: Representation, propagation, and integration," *IEEE Transaction on Aerospace and Electronic Systems* **45**, 1525–1537 (Octomber 2009).

[18] Murphy, K., Weiss, Y., and Jordan, M., "Loopy belief propagation for approximate inference:an empirical study," in [*Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*], (1999).

[19] Schrempf, O. C. and Hanebeck, U. D., "A new approach for hybrid bayesian networks using full densities," in [*Proceedings of the 6th International Workshop on Computer Science and Information Technologies (CSIT 2004)*], **1**, 32–37 (Oct. 2004).

[20] Sun, W. and Chang, K., "Unscented message passing for arbitrary continuous bayesian networks," in [*Proceedings of the 22nd AAAI Conference on Artificial Intelligence*], (July 2007).

[21] Kuo-Chu Chang, H. C. and Smith, C., "Constraint optimized weight adaptation for gaussian mixture reduction," (2010).

[22] Schrempf, O. C., Feiermann, O., and Hanebeck, U. D., "Optimal mixture approximation of the product of mixtures," in [*Proceedings of the 8th International Conference on Information Fusion (Fusion 2005)*], **1**, 85–92 (2005).